Amendments to the specification:

Please replace the sections "FIELD" and "BACKGROUND" with the following section:

BACKGROUND OF THE INVENTION

~~FIELD~~

1. Field of the Invention

This invention relates to persistent data storage techniques.

~~BACKGROUND~~

2. Description of Related Art

A large-scale database system may contain millions of records that are accessible to millions of users. Potentially, tens of thousands of data accesses on the records may take place every second. The database system may include data storage devices accessed by processes running on multiple processors. The storage devices and processors can be distributed in various locations connected via networks. For example, a large retail business could have a first storage device that maintains names and addresses of its customers, a second storage device that maintains inventory lists, and a third storage device that maintains purchasing history of its customers. The first storage device is located in Boston, the second one in Los Angeles, and the third one in Chicago. Each storage device is managed by a different processor, which is connected to the others by a wide area network (WAN). When a customer Lisa places an order for a coffee table, for example, through a clerk in a call processing center operated by the retail business, the clerk has to check, via the WAN, if the coffee table is available from the storage device in Los Angeles. The clerk may also need to access the storage devices in the other locations to retrieve Lisa's address for shipping and update her purchasing history. At the same time, another customer Robyn may place an order for the same coffee table through another clerk

in the call processing center. Both clerks will be reading from the same storage device and trying to update the same inventory record for the coffee table.

In the above example, the three different storage devices contain different types of data records that usually can be accessed independently. Using multiple processors, as in the above example, can improve the performance of the database system in terms of throughput and load-balancing, as long as data accesses are independent and each access can run on a different processor in parallel.

Because a distributed database system is accessible by multiple processes, conflicts may occur if the processes are not properly coordinated. Examples of conflicts include: two processes attempting to update the same record at the same time with two different values (as in the coffee table example); a process attempting to read a record that is being deleted by another process; and a process attempting to update a record that links to a related record being updated by another process. When a conflict happens, the operations of processes that access the same or related data records may interleave in an unpredictable way, such that the results of the operations may be incorrect and may destroy the data consistency of the database system.

One approach for resolving conflicts uses a semaphore that locks a data piece (e.g., a variable, a customer record, or a department database) when a process is accessing a data entry within the data piece, and releases the lock when the process finishes the access. All other processes must check this semaphore before accessing the data piece to see if any process is currently using it. This approach may require millions of locks on millions of data pieces if the granularity of data pieces that can be locked is small, or may block large numbers of accesses if the granularity of data pieces is large, because locking an entire department database, for example, prevents efficient parallel execution of jobs that access disjoint data sets that happen to be stored in the same department database.

In addition to conflicts, a large-scale database system may also suffer from inefficient data access. To avoid searching the entire database system just to locate a data record in a storage device, a summary information (e.g., a table of content, an index, or a cross-reference) of data records is usually provided in an easy-to search format. However, the summary information may

be subject to corruption unless its consistency with the data records is always enforced. Furthermore, the tasks of updating the summary information may also create conflicts, and therefore must be scheduled effectively.